CTS300
Prescriptive Guidance Juggling Web Services, WSE, .NET Remoting, System.EnterpriseServices, And MSMQ

Richard Turner
Program Manager
Microsoft Corporation

Connected Systems Track

Mon 10:45	CTS200 Service Orientation and the Windows/.NET Developer
Mon 1:30	**CTS300 Prescriptive Guidance-Juggling Web Services, WSE, .NET Remoting, System.EnterpriseServices, and MSMQ**
Mon 3:15	CTS308 Applied Web Services in Hewlett Packard's Core eCommerce Solutions
Mon 5:00	CTS404 Best Practices for Dealing With State at Multiple Layers Within Your .NET Applications
Tue 1:30	DEVC27 Service Orientation Overview
Tue 3:15	CTS301 Managed Programming for Win32 Developers
Wed 2:00	CTS302 Using Web Services Enhancements v2.0 (WSE) to Secure Web Services
Wed 10:15	CTS400 Using Web Services Enhancements v2.0 for Messaging Over Multiple Machines and Networks
Wed 5:30	CTS303 Applied Web Services at the Ohio State University Medical Center
Thur 8:30	CTS403 Handling Errors with Transactions
Thur 8:30	DEVC35 Web Services Interoperability
Thur 1:30	CTS406 Versioning of Connected .NET Applications
Thur 1:30	DEVC16 Distributed Applications Performance Panel
Thur 5:00	CTS304 Availability and Reliability - Failure Is Not An Option
Fri 10:45	CTS405 Choosing a Hosting Model - Dealing with Threads, AppDomains, and Processes
Fri 1:00	CTS306 Deployment and Management
Fri 2:45	CTS307 Using Service Orientation to Drive Business Processes

What We'll Be Discussing

- What is Service Orientation?
  - What's all the fuss about?
  - Why is it important?
- How do I build SO Applications today?
  - Where to most appropriately use existing technologies
  - Where NOT to use existing technologies …And why!
- System Performance

What Is Service Orientation (SO)?

- SO is a paradigm, an approach, NOT a technology or product
- An approach to building Systems using Services which adhere to …
- The 4 Tenets of Service Orientation
  - Boundaries are Explicit
  - Services are Autonomous
  - Services share Schema and Contract, not Class
  - Service compatibility is determined based on Policy

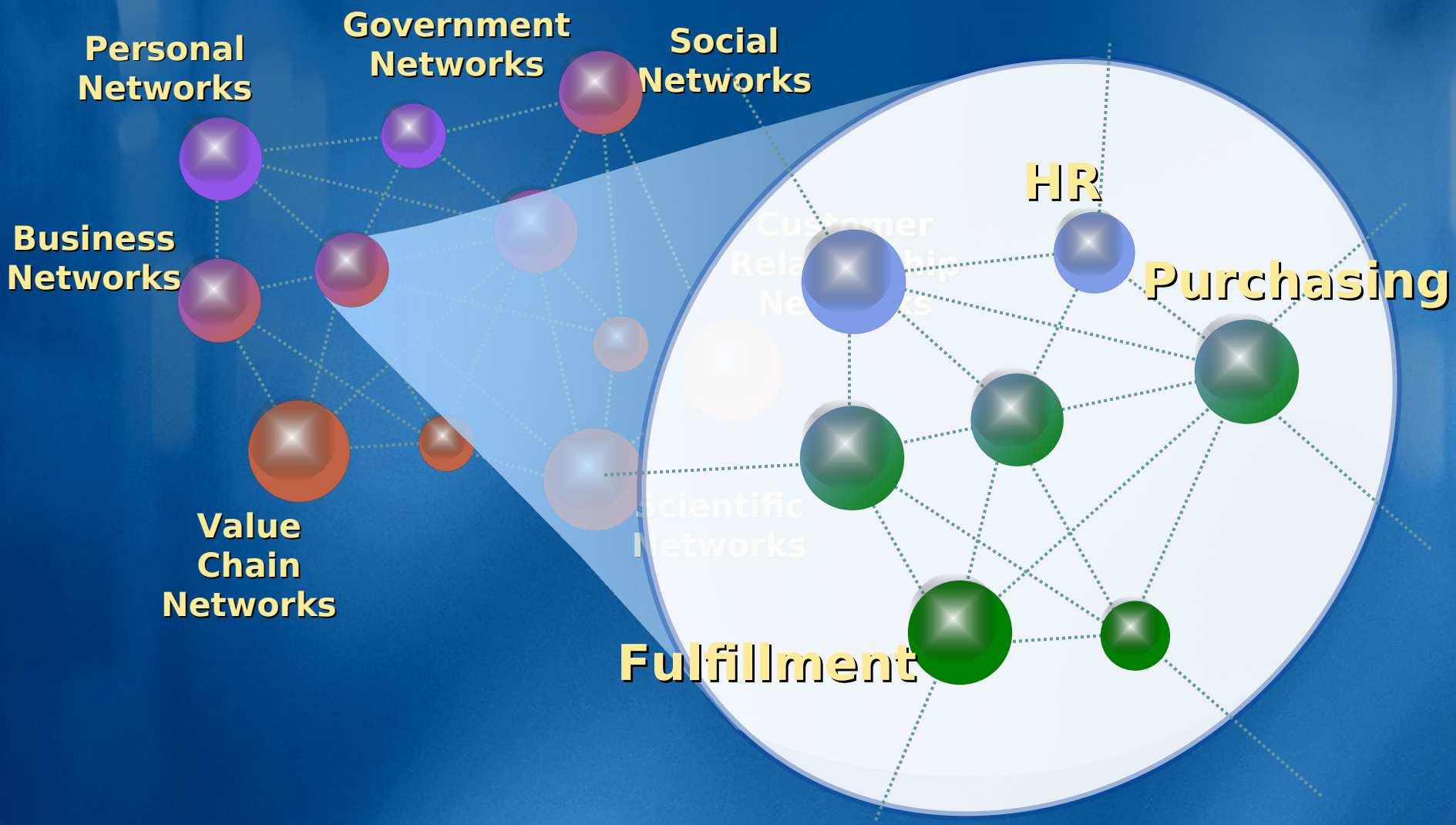
http://msdn.microsoft.com/msdnmag/issues/04/01/Indigo/default.aspx

Services And Systems

- A Service is an entity you interact with via message exchanges
  - Services are built to last
  - Availability and stability are critical
- A System is a set of deployed Services cooperating in a given task
  - Systems are built to change
  - Adapt to new services after deployment
- Services are Fractals
  - A service can be composed of sub-services…
  - …Which can be composed of sub-services…

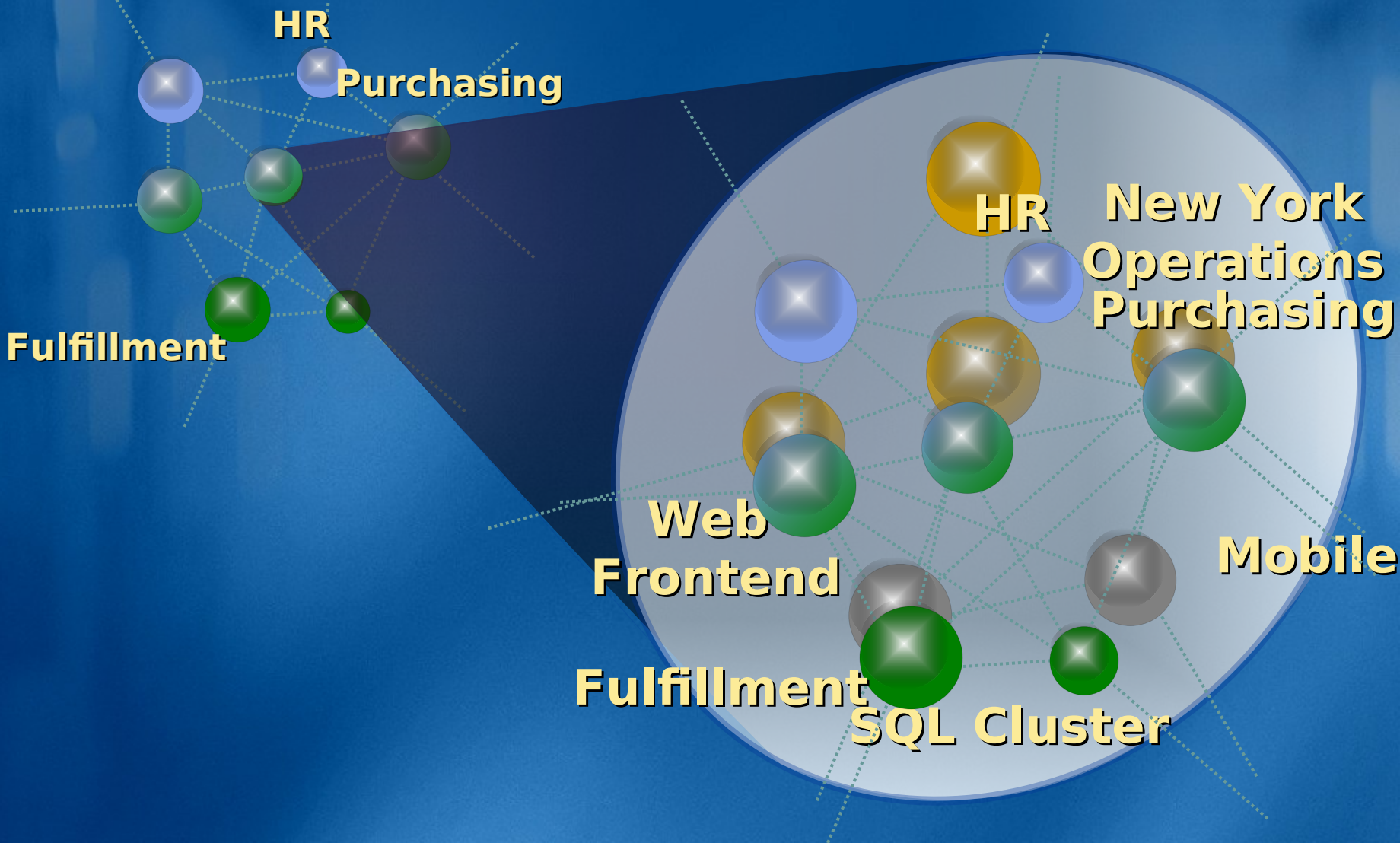
The Microsoft Bet

Services Everywhere



The Microsoft Bet

Services Everywhere



The Role Of Objects

Object Orientation

Interaction Richness

- Complete fidelity with local contract
  - Customers have 20+ years of experience and have built intuition here
- Boundaries are harder to isolate
  - Requires careful opting out
- Share types, not schemas
- Tightly Coupled
  - Typically require deployment of both client and server in sync
  - Assume cheap, transparent communication
  - object identity and lifetime maintained by infrastructure
- Platform-based compatibility
  - Assertions built into platform

Service Orientation

Reduced Assumptions

- Reduce assumptions between services
  - Builds on ideas from component software, distributed objects, and MOM
- Boundaries are explicit
  - Opt in model
- Share schemas, not types
- Autonomous
  - Security and failure isolation are a must
  - Variable cost, explicit communication
  - Independent deployment of client/Service
- Policy-based compatibility
  - Assertions use stable global names

Why Is SO Important?

- Provides us an opportunity to rethink how we design and architect tomorrow's systems
  - Minimizing hard interdependencies
  - Enhancing independence
  - Easing delivery of composable business apps
- Enables a high level of interoperability
- Makes explicit some of the core assumptions that compromise today's systems
  - Particularly around boundaries and

Benefits Of SO

- Architecture and Development
  - Formal interaction model facilitates simplicity, correctness, implementation independence, dependency management
- IT and Operations
  - Explicit interaction points are more discoverable, operable
  - Crisply isolates service capability from IT environment
  - Independent deployment, versioning, management, topology
- Business
  - Services promote technology reuse, resulting in cost savings
  - Services model business capabilities
  - Systems serve the business, not vice versa
  - Inter-departmental or inter-org relationships formalized and expressed through service interaction
  - Facilitates outsourcing and focus on core competencies

Practicalities

- A common tongue is needed for services to interact
  - Boundary, schema, contract, policy
- A SO environment extends only as far as we agree on the expression of the boundary
  - How far do you want your boundary to extend?
- SO Systems that want broadest possible interoperability will build on the WS-* protocol family
- Microsoft offers great tools for

Building Service-Oriented Applications

Using today's technologies

Design With SO In Mind

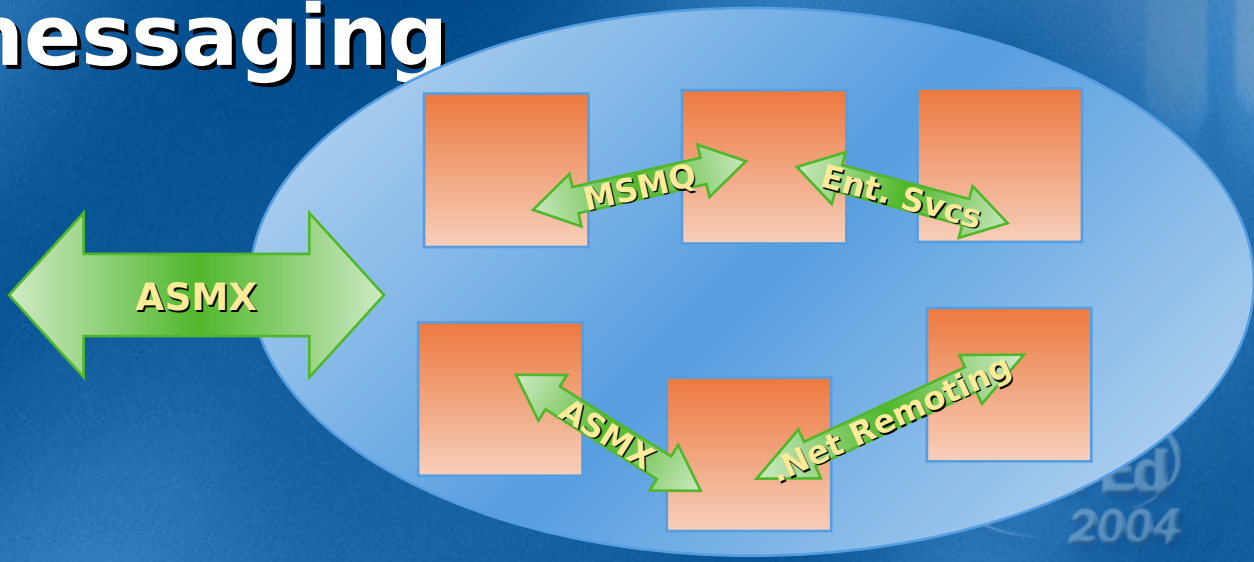
- Build SO systems today!
  - Improve your chances of interop
  - Increase autonomy within your systems
  - Resultant systems are more resilient
  - Easier to version
  - Migration to “Indigo” will be simplified
- Building fully SO systems today can be tricky

Implementing SO today

- Adopt open standards at service boundaries
  - Standard transports: HTTP, TCP
  - Standard data formats: XML, XSD, SOAP
  - Standard protocols: WS-Security, WS-Addressing, WS-Policy, WS-*, …

Which Technology Should I Use And Where?

- Build services using ASMX
- Components should stay WITHIN your Service boundaries
- Use System.Messaging for async queued messaging



At The Service Boundary

- Use ASMX at the Service Boundary
  - Closely aligned to SO Tenets
  - Closest alignment with “Indigo”
  - Great interop support
- Use WSE for advanced Web Services
  - Support for WS-* protocols
  - Interop with systems that demand WS-* support
- Great scale-out capabilities

Inside the Service Boundary

- Consider ASMX inside Service boundary too!
- Use Enterprise Services (ES) if…
  - You need its rich services
  - Want to re-use/extend existing ES/COM+ components
- We're doing a lot of work to build a path to "Indigo" from ES

Asynchronous Comms

- Use System.Messaging if you need
  - Asynchronous messaging
  - Reliable messaging and queuing
  - “Fire and Forget” messaging
- Note: System.Messaging namespace not carried forward to “Indigo”
  - “Indigo” natively supports Queueing semantics

But What About Remoting?

- Use .NET Remoting when appropriate
  - Only inside the Service!
  - Inproc X-AppDomain comms
  - Handling custom wire protocols
- Avoid exposing Remoted components at Service Boundaries
  - Remoting is an object technology – not aligned with SO principles
  - Limited interop

Today's Caveats

- ASMX
  - Avoid or abstract using low-level extensibility such as the HTTP Context object
- Enterprise Services
  - Avoid passing object references inside of ES
- Native COM+ and MSMQ
  - Use .NET System.EnterpriseServices and System.Messaging – do not use native COM+ and MSMQ APIs
- .NET Remoting
  - Avoid or abstract using low-level

Distributed Technology Performance

Enterprise Services, Remoting, and ASMX

ASMX Performance

- Customer: "Won't my app slow down if I convert it to an ASMX fronted service?"
- Not necessarily…what % of your method calls are work?
  - More work than call – probably minimal impact
  - More call than work – possible impact
- General guidelines
  - Only call when you need to
  - When you do, perform lots of work
  - Only pass what you need on the wire
- These are immutable guidelines for all

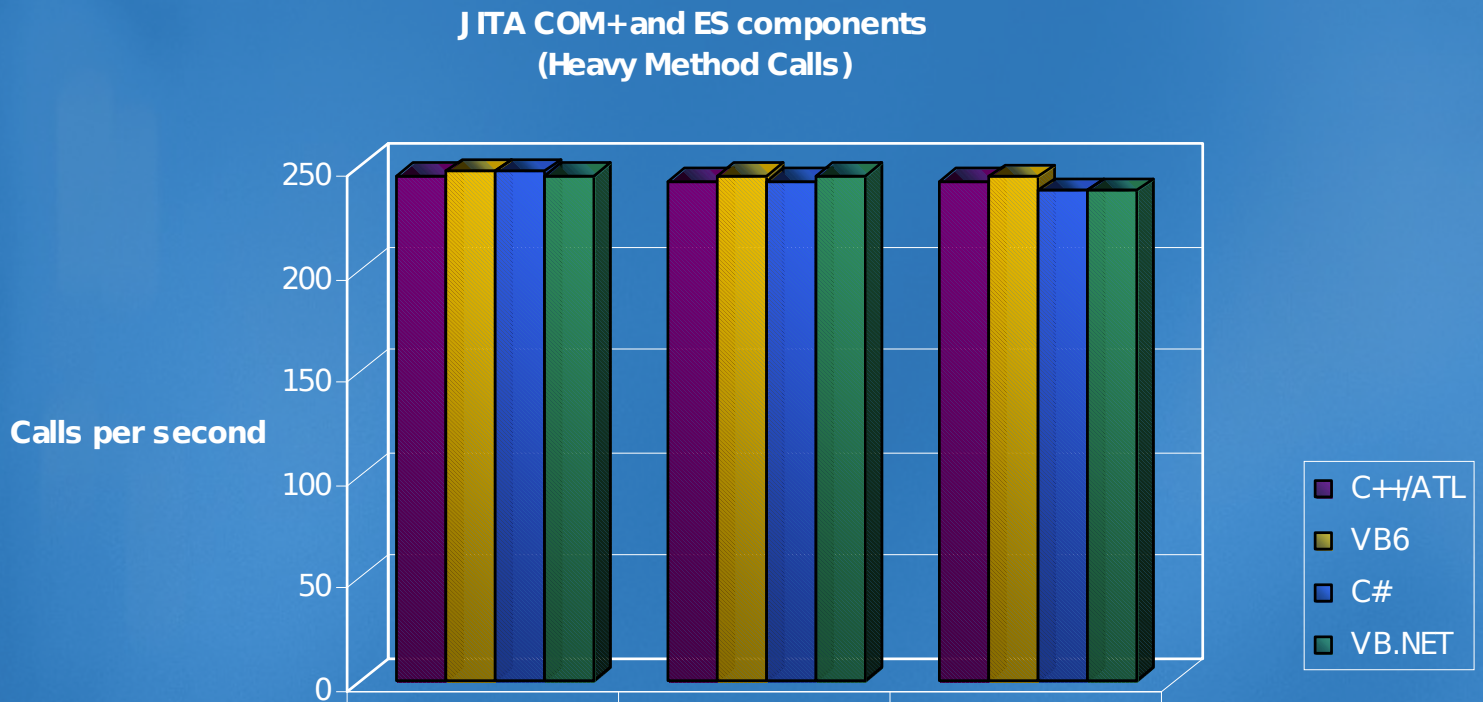
Enterprise Services Performance

- Customer: "Will my code slow down significantly if I port my components to ES?"
- Not if you follow our guidelines
- Use JIT Activation (JITA) and Object Pooling
  - Amortize cost of instantiation over many method calls
- Make methods perform substantial work
- Minimize number of method calls required on COM+/ES components

COM+ Versus ES Performance

- Gotchas to be aware of:
  - ES objects require more work to create and destroy than COM+ objects
    - In .NET 1.x - 5 calls to activate ES versus 3 for COM+
    - Resolved in .NET 2.0, but ES components still incur overhead over COM+
  - Always explicitly dispose of ES objects when finished
- If used effectively, ES can match COM+ in performance!
  - Plan to use JIT Activation & Pooling to amortize creation and destruction cost over many calls
- See “Enterprise Services Performance” paper on MSDN:

ES CAN Match COM+ Perf!



	In Process	Out Of Process	Cross Machine
C++/ATL	245.4	243.3	242.9
VB6	247.6	245.4	245.2
C#	247.6	243.3	238.5
VB.NET	245.4	245.4	238.5

http://msdn.microsoft.com/library/en-us/dncomser/html/

Remoting Versus ASMX Performance

- See Ingo Rammer's perf comparison paper (www.ingorammer.com)
- Results are as follows
  - TCP-Channel Binary Remoting is the quickest
  - Binary Remoting of component hosted in IIS is generally next fastest
  - ASMX is a little faster than Remoting SOAP over HTTP!

Performance Guidelines

- Good practices apply across all distributed technologies!
  - Minimize NW traversals
  - Chunky interfaces == better perf than chatty
- Component technologies often obscure boundaries
  - Design/dev assumptions do not equate to deployment-time realities
  - Resulting in compromised perf
- SO systems make up-front assumptions about boundaries
  - Usually results in more pragmatic designs and implementations
  - Usually deliver good performance, despite traffic overheads!

Summary

- Service Orientation supports the building of autonomous, resilient, dynamic, agile Connected Systems
- Build SO systems today!
  - Use .NET technologies appropriately for maximum effect
  - High-performance CAN be achieved
  - You will benefit from more dynamic, agile, composable systems
- For more info and other links on this subject, visit
  - http://blogs.msdn.com/richturner666

Session Evaluation

Please fill out a session evaluation on CommNet

Q1: **Overall satisfaction with the session**

Q2: **Usefulness of the information**

Q3: **Presenter's knowledge of the subject**

Q4: **Presenter's presentation skills**

Q5: **Effectiveness of the presentation**

Community Resources

Attend a free chat or web cast
http://www.microsoft.com/communities/chats/default.mspx
http://www.microsoft.com/usa/webcasts/default.asp

List of newsgroups
http://communities2.microsoft.com/communities/newsgroups/en-us/default.aspx

MS Community Sites
http://www.microsoft.com/communities/default.mspx

Locate Local User Groups
http://www.microsoft.com/communities/usergroups/default.mspx

Community sites
http://www.microsoft.com/communities/related/default.mspx

Microsoft®

Your potential. Our passion.™

Microsoft Tech·Ed 2004